Random positions in Go

Benard Helmstetter, Chang-Shing Lee, Fabien Teytaud, Olivier Teytaud, Wang Mei-Hui, Shi-Jim Yen

Random positions in Go

Bernard Helmstetter, Chang-Shing Lee, Fabien Teytaud, Olivier Teytaud, Mei-Hui Wang, Shi-Jim Yen

***Abstract*—It is known that in chess, random positions are harder to memorize for humans. We here reproduce these experiments in the Asian game of Go, in which computers are much weaker than humans. We survey families of positions, discussing the relative strength of humans and computers, and then experiment random positions. The result is that computers are at the best amateur level for random positions. We also provide a protocol for generating interesting random positions (avoiding unfair situations).**

I. Introduction

Computers are much stronger than in the past in Go. Thanks to milestones like Monte-Carlo Tree Search[1], Rapid Action Value Estimates[2], patterns[3], combinations of patterns and RAVE values [4], parallelization[5], [6], opening books[7], computers became much stronger, especially in the 9x9 board. Nonetheless, humans are still stronger, by far, in the 19x19 board; the best performances so far are with H6 against pros and H7 against top pros. H6 and H7 means 6 and 7 stones of handicap. Having X stones of handicap means that you can play X stones on the board before your opponent can start to play.

In chess, it is known that expert players (or really strong amateurs) are able to memorize positions shown only for a few seconds with almost no errors. When the positions are random, expert players are not able to recall the positions with less errors than amateur players (or not much less). Then, it is the recognition of patterns which are memorized. More on these experiments can be found in [8], [9], [10]. It is known that in Go, positions on which computers make stupid mistakes include semeais, or sophisticated life & death problems, involving a high level of abstraction[11]. Random positions might make things very different, as such situations might not occur.

Interestingly, the fact that Go starts from an empty board is not a constant:

- in Tibetan Go, there are, initially, 12 stones on the board (see Fig. 1). In this version of Go, the rules of ko and snapback are also significantly modified, as well as the territory system.
- in Sunjang Baduk (Korean variant) there are 16 stones (see Fig. 1), and the first black move is fixed at the center (leading to 17 initial stones and white starting).
- in Bantoo (Korean version played mostly with computers), each player places three stones; there are other significant differences (including the possibility of playing a hidden stone once in the game, and scoring differences).

S.-J. Yen is with the National Dong Hwa University, Taiwan; F. Teytaud and O. Teytaud are with TAO (Inria), LRI, UMR Cnrs 8623 Univ. Paris-Sud, C.-S. Lee and M.-H. Wang are with the National University of Tainan, Taiwan. B. Helmstetter is with the LIASD, Université Paris 8.

However, the board is always "almost" empty in the sense that there is enough room for building classical figures. Also, it is sometimes said in Go that the fact that Go moves from an empty board to a full board is in the spirit of the game (this is not the case in chess, draughts, checkers,...); more philosophical elements around that in [12] (chapter devoted to Go).

Section II discusses random positions in other games. Section III discusses other families of positions in Go. Section IV discusses situations in which computers are very weak. Section V presents games between a strong amateur player and a MCTS program from random positions. Section VI presents a Go-expert point of view on random positions.

II. Random positions in chess and other games

In Chess, Bobby Fischer created in 1996 a new variant of the game of Chess, called Fischer Random Chess (also termed Chess960). In this variant, rules are similar to the game of Chess, except that the initial position is interestingly modified. Each white piece is randomly placed on the board, with respect to some constraints, for instance, pawns are on the second rank (as in Chess), Kings have to be between the two rooks and the two bishops are placed on opposite color locations. Black pieces are placed by symmetry. The motivation of Bobby Fischer was to keep the game of Chess as tactical and strategical as possible. In Chess, a lot of opening theory exists, and the beginning of the game is often played "by heart". Randomly moving the initial position avoids this opening preparation.

Figure 2 presents three of the 960 possible initial positions.

The resulting rules of this variant are indeed more complex than the Chess rules. As said previously, constraints exist on the initial position, but there are also specific rules for castling for instance.

This variant is the most famous chess variant. Several top players play also in the World Chess9600 championship. It is interesting to note that top players in Chess are generally good in 960Chess (but the winner of both championships are not the same players). Unfortunately, not so many games between humans and computers have been played to be able to extract some results.

III. Other families of situations in Go

To the best of our knowledge, random positions have not been much investigated in Go. Other important families of situations are as follows:

- Ladders (shishos): ladders are well known for being PSPACE complete; [13] provided a go position which encodes a quantified Boolean formula (Fig. 5).

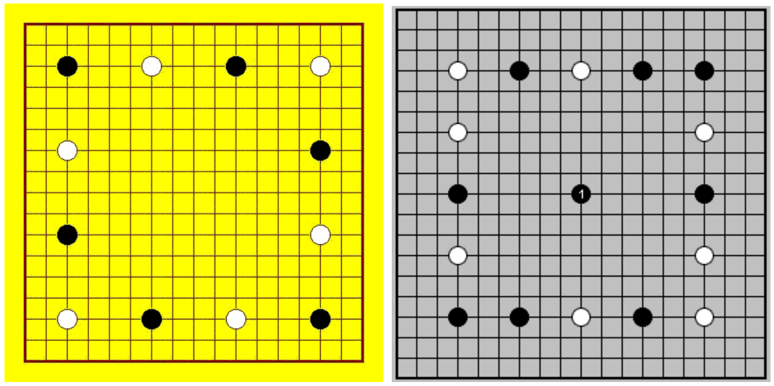

Fig. 1. Left: Initial position in Tibetan Go (source:Wikipedia).Right: Initial position in Sunjang Baduk Go (source:Wikipedia).

- Ko-fights: ko-fights are very important in Go, and disappear in some variants (for beginners) like Ponnuki-Go; they are the crucial component in the EXPTIME-completeness proof of Go with japanese rules (with no superko). Unfortunately, the problems given in [14] are so big that they can not be drawn in an article or tested by humans. It is known that computers sometimes make mistakes in ko-fights, by wasting threats.
- Semeais (liberty races): semeais become famous in computer-Go as they have been shown as critical in computer vs computer games (because they are often badly and randomly played by computers, leading to unpredictable results) and in games vs humans (because humans often win by such situations). For example, MoGoTW won the TAAI 2010 competition in 19x19 by winning against DeepZen (the cluster version of the Zen program) thanks to a misreading by Zen, switching from a clearly won situation to a clearly lost situation, without any of the two bots having clearly understood what happens (Fig. 6).
- There are also Nakade, i.e. cases in which a player kills an opponent by playing some stones inside his opponent's group (Nakade were known as a main weakness in computer-go, but at least simple nakade are now solved by special tricks in Monte-Carlo Tree Search - however, not all nakade are well handled (see Fig. 3 for some examples.
- Ishi-no-shitas are complex situations involving captures and recaptures inside a group; it is known that in such situations (difficult to visualize as everything occurs "under" initial stones), computers are often stronger than humans.

All these complicated situations occur by the combination of two players constructing something meaningful. We will consider in the rest of this paper random initial positions.

IV. Situations on which computers are weak

Figure 7 (derived from [11]) shows a simple semeai which is very poorly analyzed by computers. Even a beginner would play correctly these situations: in Fig. 7 (left), the semeai is urgent as the number of liberties is the same for both players; whereas in Fig. 7 (right), the semeai is not urgent as (black as more liberties) and black should play somewhere else. We point out that this weakness is not only a property of MoGo; it has been reported as a weakness in the computer-go mailing list for all current implementations of Monte-Carlo Tree Search. These situations do not necessarily occur in random games; we will investigate this in section V.

V. Games against a 5D player

The protocol is explained in section V-A; basically, games are randomly generated, non-equilibrated situations are discarded, and the human benefits from a pie rule (i.e. can choose between black and white) so that the situation can not be in favor of the computer (at least, not in an obvious manner) - in case of perfect play (including perfect choice between black and white) the human should win everything. Results are then presented in section V-B.

A. How to generate and use random games

We generated random games as explained in Alg. 1. The

Algorithm 1 Algorithm for generating a random position with N stones.

ok← false
while not ok **do**
  Randomly put $N/2$ black stones and $N/2$ white stones on the board.
  if situation is legal **then**
    Play 50 games by a MCTS algorithm with 10 000 simulations per move from this situation
    if number of wins for black $\in [22, 28]$ **then**
      ok← true
    end if
  end if
end while

protocol used for playing against humans is then as explained in Alg. 2. The program was MoGo, running on a 16-cores 2.96GHz machine.

Algorithm 2 Protocol used for human vs computer game from random position.

Pick up a random position as generated by Alg. 1.
The human has 3 minutes for analyzing the position and deciding between black and white.
Play the game:
Humans has 30 minutes + 30 seconds per move, computer has 30 seconds per move.
if Human wants to try with other side **then**
  Play the same situation with colors exchanged.
end if

B. Results

The human player is Bernard Helmstetter (BH). Bernard Helmstetter, born in 1977, is French 5Dan, french champion in 2003, ranked 4th in the world amateur championship 2004. He is also a computer scientist, has experience in Monte-Carlo Go and is a particularly difficult opponent for computers, usually winning with handicap 6 against MoGo. As a summary, BH won with 50, 70, 100, 160 random stones; also another position with 128 stones distributed following a proposal by BH. With 180 stones, there was a first position in which MoGo won both as white and black; another position with 180 stones was tested, and BH won this one. A test with 240 stones was a win for white each time (for MoGo and then for BH).

Section V-B1 presents results with small numbers of random stones; Section V-B2 presents results with big numbers of random stones.

1) Games with at most 160 random stones: BH won (not always easily; the situation in the games with 70 stones was difficult for the first moves, but BH won easily after MoGoTW let him take and connect the ko) games with 50, 70, 100, 128 and 160 random stones respectively. The situation with 70 stones is shown in Fig. 8. Fig. 9 shows that with 50 stones, we can have clear semeais; MoGo misread the huge semeai on the left and lost the game (we played the game until MoGo understood the result, but it was clear very early for humans that the result was a win for the human).

2) Games with more than 160 random stones: Then, BH lost a game with 240 stones (Fig. 10), in which he had chosen black (without really checking the position as there was no obvious advantage on the figure). However, he then tried again with white and won quickly (MoGo resigned after 12 moves). This suggests that the situation was easier for white.

We then tried again with 180 random stones. The first tested situation is shown in Fig. 11 (left). The computer won as black (Fig. 12, left), and then again as white (Fig. 12, right), suggesting that the situation is equilibrated and better understood by MoGo than by the human.

We then tried another situation with 180 stones (Fig. 11, right), for which the human chose black and won.

VI. Assessment of the difficulties of random Go positions

The game results show that the superiority of the human player is reduced when starting from random positions with many stones. We shall discuss the reasons. Many aspects of the game change in random positions; how they change is influenced by the number of stones; these changes impact human players and computers differently. Finally, each random intial position, even with a fixed number of stones, has its own flavour. In this section, we investigate the changes that happen in several aspects of the game, some of which may overlap others.

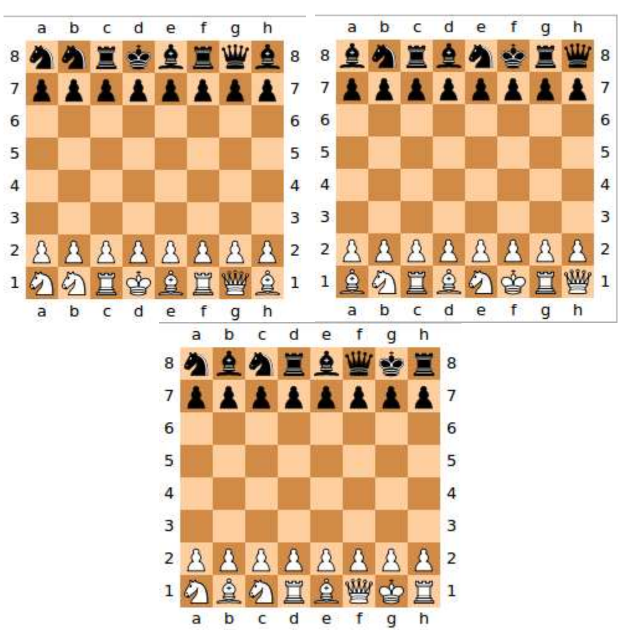

Fig. 2. Three of the 960 possible initial positions. In the first initial position (top-left), the a and b pawns are not protected and can be directly under attack if either the f or the g pawns of the opponent are played.

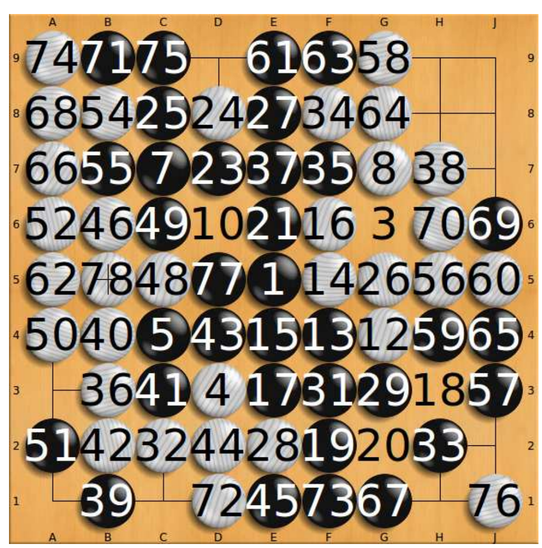

Fig. 3. MoGoTW played against Jujo Jiang (9P) in 9x9; he won one out of two games as black, and lost one game as white. In the presented game, we see that MoGoTW (white) lost this third game, because it did not understand the nakade (bottom left), whereas it is extremely simple. The program can therefore beat pros (see Fig. 4), but make huge mistakes like this one.

A. Local tactics and shapes

The shapes in random positions are often unusual and disturbing to the human player. This is particularly true near the edges, with some random stones lying isolated on the first line, which generally doesn't happen in normal games until the endgame. This disadvandage to the human player, while real, is not so important, and would probably be reduced with some training. BH thinks that he still has some advantage over MoGo in this area. However, the computer is strong at killing big groups by chasing them towards the center.

B. Global strategy and stability

To the human player, random positions are also unusual at the global level. The situation is however very different from one position to another, depending primarily on the number of initial random stones, but also on the existence of stable groups. With many stones, the games will start directly in the endgame, rather than the middlegame. The global aspect will be reduced or absent. The global analysis is typically complicated by the many unstable groups lying on the board, the strengths being more difficult to estimate. Group sacrifices are more frequent as the game advances. It can be noted, however, that the positions are often not as difficult to the human player than they could be, because the random generation often produces a few very strong groups, the existence of which considerably simplify the analysis. BH thinks that he still has a big advantage over MoGo in the area of global strategy. The presence or absence of this global dimension in a random position might be the main factor to consider in order to explain why the position would be more or less advantageous to the computer. The first 180 stones random position that was won by MoGo with both sides can be classified as a late middlegame position. The global dimension is not absent but is much reduced: it essentially only concerns the center area. There is some unstability on the left side, and the lower left and lower right corners, but they do not have whole board consequences. In some sense, although the board is 19x19, its effective size is less, since many areas (upper left corner, upper right corner, right side, and left side) are very stable. To summarize, the amount of global unstability on the board is a factor of some confusion to the human player, but probably has a stronger negative impact on the computer's playing strength.

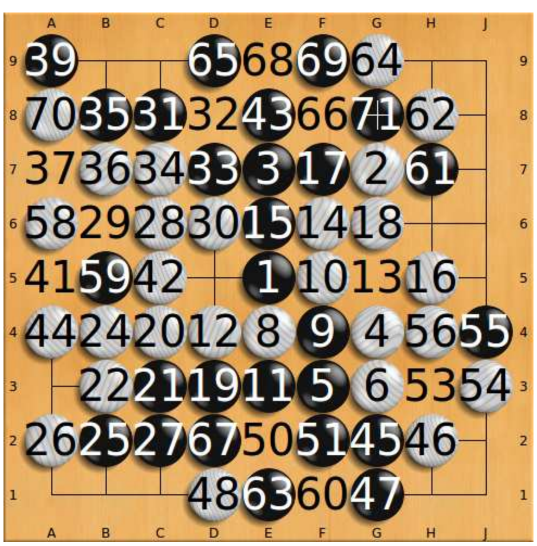

Fig. 4. MoGoTW won this game as black (the difficult side, with komi 7.5) against Jujo Jiang (9P) in 9x9 (10seconds/move for the computer; 16 cores, 3GHz; no time limit for the human).

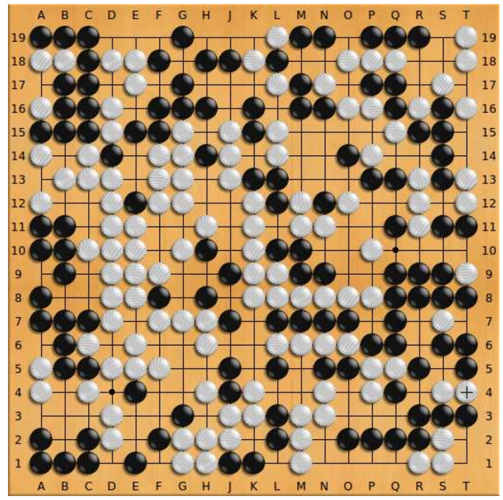

Fig. 10. The random position with 240 stones; the computer won as white and then the human won as white, suggesting that (maybe) this situation is easier for white.

C. Games that begin in the early endgame

Unlike games with less initial random stones, some games can be said to start directly in the early endgame, which is quite different to the middle game. The global aspect is much reduced in the endgame; the analysis of an endgame position can generally be broken into subproblems with little dependence between them. The position with 240 stones and the second game with 180 stones fall into this category. The 240 stones one was however more unstable and difficult to analyze. MoGo's endgame skills are generally good (although he has been seen to blunder and lose winning games during some training games from non-random positions). A position like the 240 stones one offers some advantage to MoGo in that he can accurately estimate the positions, while the human player has difficulties counting them, especially under tight time constraints.

D. Life and death in the corners

One of the weaknesses of MoGo in normal games, and especially in handicap games with initial corner stones on 4-4 points, lies in securing the corners. MoGo has problems with some of the life and death problems that can arise there, especially those involving big eye spaces (nakade). Those problems are well known to good human players. Also, MoGo generally has a center-oriented style with a somewhat excessive disregard for corner territories. In random games, with some stones already in the corners, this weakness of MoGo showed less.

E. Ko fights and semeais

MCTS algorithms are known to be weak in ko fights and semeais. There have been no significant ones in the games won by MoGo, and this might have been one of the reasons for the wins. We can however think of no particular reason why ko fights or semeais would happen less frequently in games from random positions, even with many initial stones.

VII. Conclusions

We checked that randomly generated Go positions are much more difficult to analyze by Go players (compared to computers) than usual positions. This is consistent with results in chess. We generated equilibrated random situations; this involved checking fairness and rejecting desequilibrated situations. Whereas with small numbers of random stones (section V-B1), humans win easily, in particular with better skills than computers in ko-fights (Fig. 9) and semeais (Fig. 8) and life-and-death problems, the situation becomes more equilibrated with more stones (section V-B2).

Computers seemingly become competitive at the highest amateur level at around 180 random stones generated as shown in Alg. 1. We point out that in all games, the human player benefited from a pie rule (with too limited time settings for a very good choice; the human essentially used black as first choice except in the first "180 stones" situations in which the human believed erroneously that the situation was a clear win for white), and that in one case, the computer won with both sides. On the other hand, the human player played many games in the same day, which makes it hard for him to be at his best level. Nonetheless it is very clear that MoGo would never win a game against a human with no handicap from the empty board; such a result is only possible from non-standard positions. We have seen that even with pie rule in favor of the human (pie rule with limited time however, only aimed at avoiding situations clearly in favor of the computer), the computer can win (see result with 240 stones), and sometimes the computer can win with both sides; this suggests that, as in chess, humans are highly specialized on a small subset of the set of possible boards, which are those reachable from an empty board. Humans have built the ability to solve semeais and complicated life and death situations, which are crucial in usual games (with empty initial board). These conclusions are for sure extracted from a quite small sample of games; yet, a win in 19x19 with no handicap and with both sides suggests that there is a big difference with standard Go conditions. Incidentally, maybe programs strong at playing random-go are different from programs strong at playing standard go; obviously, openings make no sense, but maybe also patterns should be handled differently.

Last but not least: generating funny games. We point out that games with a random component are much more likely to motivate children (we tested this briefly on two children; more extensive experiments are required but we do not have a lot of doubts on it). Therefore, random initial positions (or maybe games with random stones added during the games) might be interesting for pedagogical reasons: it makes the game simpler, funnier; also, it is compliant with handicap stones. The algorithm we use for generating fair random positions can also be used for generating strong initial position for a weak player. Also, random positions are interesting for diversifying small boards: professional players often start small board games with a strange opening for making the game more fun and our tool makes exactly the same. Importantly, as for Fisher's random chess, we get rid of the task of learning complicated fuseki, making the game more diversified and less tedious.

Acknowledgements: We are grateful to the Grid5000 project for supporting MoGo and its parallelization since its early times. The authors would like to thank the National Science Council of Taiwan for financially supporting this international cooperation research project under the grant NSC 99-2923-E-024-003-MY3. Additionally, this work was supported by the French National Research Agency (ANR) through COSINUS program (project EXPLO-RA ANR-08-COSI-004).

References

[1] R. Coulom, "Efficient Selectivity and Backup Operators in Monte-Carlo Tree Search," *In P. Ciancarini and H. J. van den Herik, editors, Proceedings of the 5th International Conference on Computers and Games, Turin, Italy*, pp. 72–83, 2006.

[2] S. Gelly and D. Silver, "Combining online and offline knowledge in UCT," in *ICML '07: Proceedings of the 24th international conference on Machine learning*. New York, NY, USA: ACM Press, 2007, pp. 273–280.

[3] B. Bouzy and G. Chaslot, "Bayesian generation and integration of k-nearest-neighbor patterns for 19x19 go," *In G. Kendall and Simon Lucas, editors, IEEE 2005 Symposium on Computational Intelligence in Games, Colchester, UK*, pp. 176–181, 2005.

[4] A. Rimmel, O. Teytaud, C.-S. Lee, S.-J. Yen, M.-H. Wang, and S.-R. Tsai, "Current Frontiers in Computer Go," *IEEE Transactions on Computational Intelligence and Artificial Intelligence in Games*, p. in press, 2010. [Online]. Available: http://hal.inria.fr/inria-00544622/en/

[5] T. Cazenave and N. Jouandeau, "On the parallelization of UCT," in *Proceedings of CGW07*, 2007, pp. 93–101.

[6] S. Gelly, J. B. Hoock, A. Rimmel, O. Teytaud, and Y. Kalemkarian, "The parallelization of Monte-Carlo planning," in *Proceedings of the International Conference on Informatics in Control, Automation and Robotics (ICINCO 2008)*, 2008, pp. 198–203.

[7] P. Audouard, G. Chaslot, J.-B. Hoock, J. Perez, A. Rimmel, and O. Teytaud, "Grid coevolution for adaptive simulations; application to the building of opening books in the game of Go," in *Proceedings of EvoGames*. Springer, 2009, pp. 323–332.

[8] A. De Groot, *Thought and choice in chess*. Mouton De Gruyter, 1978.

[9] A. De Groot, F. Gobet, and R. Jongman, *Perception and memory in chess: Studies in the heuristics of the professional eye.* Van Gorcum & Co, 1996.

[10] F. Gobet and H.-A. Simon, "Recall of random and distorted positions: Implications for the theory of expertise," *Memory & Cognition*, no. 24, pp. 493–503, 1996.

[11] A. Bourki, G. Chaslot, M. Coulm, V. Danjean, H. Doghmen, J.-B. Hoock, T. Hérault, A. Rimmel, F. Teytaud, O. Teytaud, P. Vayssière, and Z. Yu, "Scalability and parallelization of monte-carlo tree search," in *Proceedings of Advances in Computer Games 13*, 2010.

[12] K. Chairasmisak, *Enseignements d'un dirigeant asiatique (exists in Chinese)*. D'organisation, 2005.

[13] D. Lichtenstein and M. Sipser, "Go is polynomial-space hard," *J. ACM*, vol. 27, no. 2, pp. 393–401, 1980.

[14] J. M. Robson, "The complexity of go," in *IFIP Congress*, 1983, pp. 413–417.

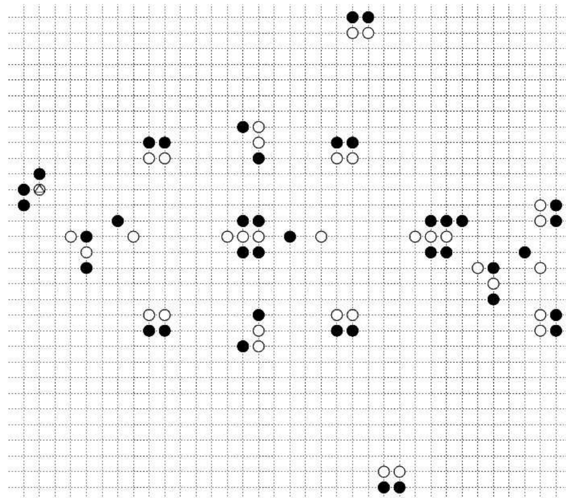

Fig. 5. A ladder (can black capture the Δ stone ?) which encodes a quantified Boolean formula.

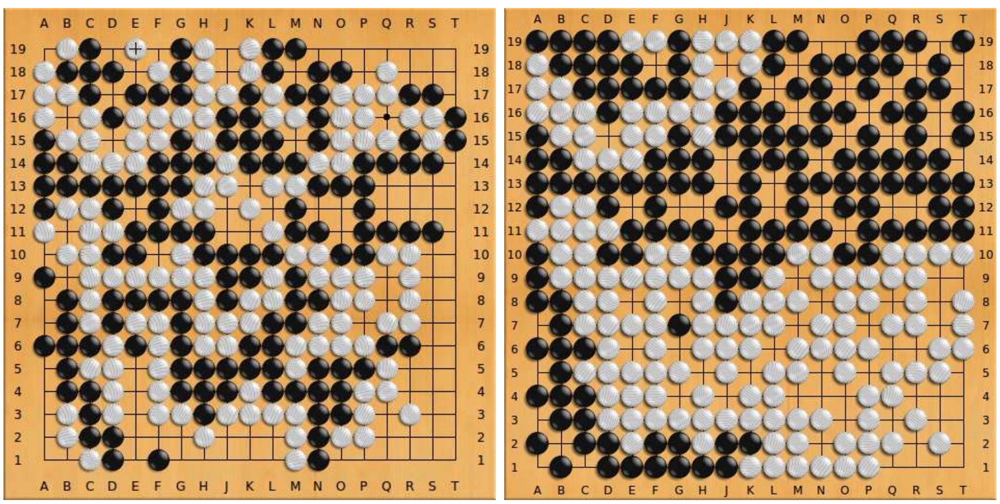

Fig. 6. Left: the situation in this game (MoGoTW vs DeepZen, TAAI 2010) was a win for DeepZen (black; the cluster version of Zen) until DeepZen made a big mistake leading to a seki (top left part). Right: the final situation (seki, the white groups in the top left are alive) .

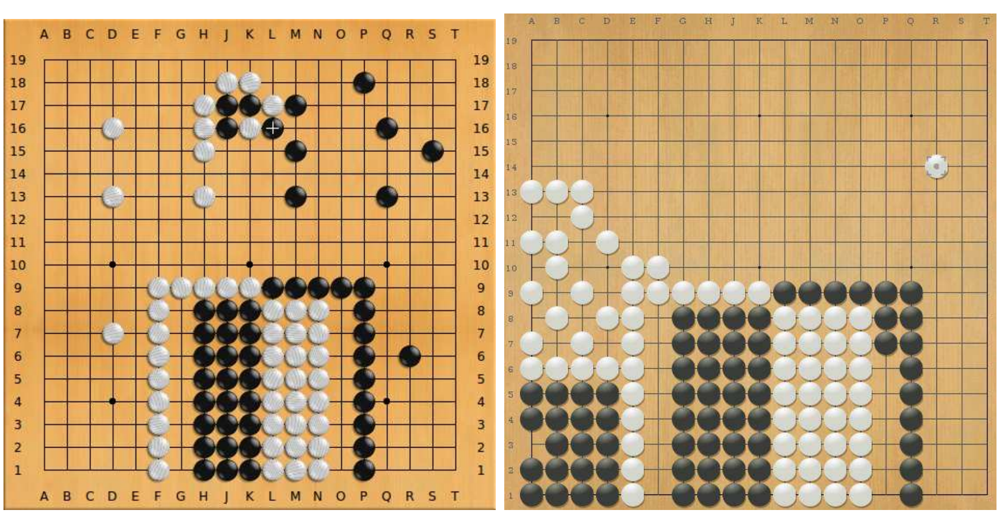

Fig. 7. Left: the semeai is urgent and should be played now. Right: the semeai is not urgent and should not be played. Computers usually don't evaluate these situations correctly.

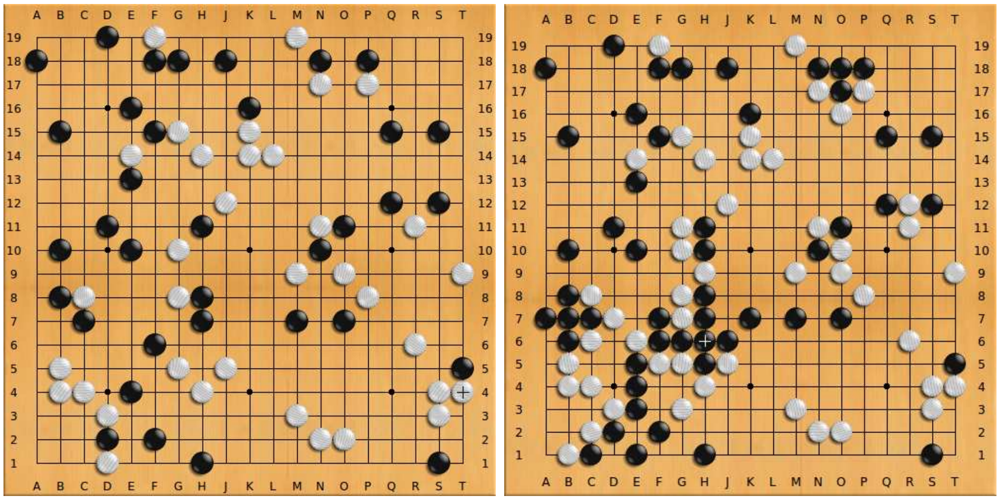

Fig. 8. Situation with 70 random stones (left); the situation was difficult at the beginning, but BH won easily after MoGoTW let him take and connect the ko (right: human (black) plays H6 and connects the ko).

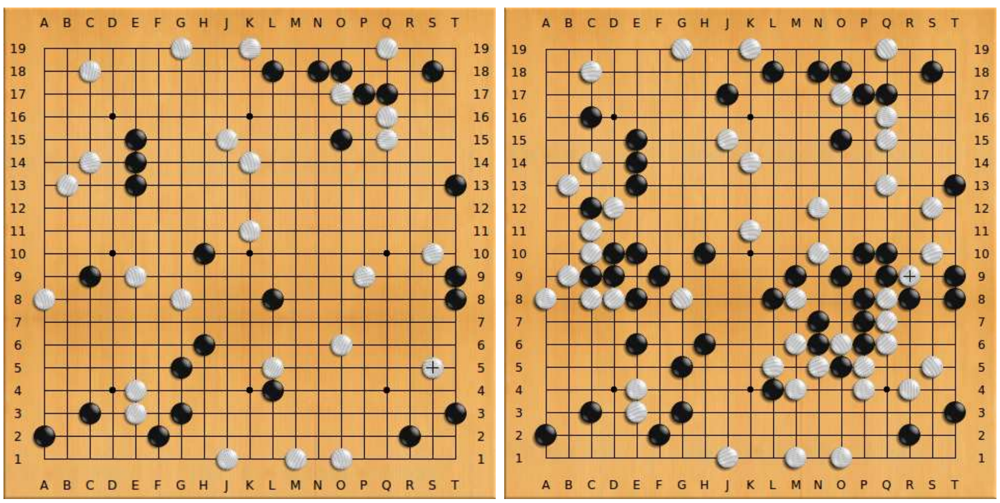

Fig. 9. Situation with 50 random stones; BH won easily. The position on the right shows some unfamiliar shapes that random go positions can lead to. If black tried to capture the cutting stone, the first line stones would get involved in unusual ways.

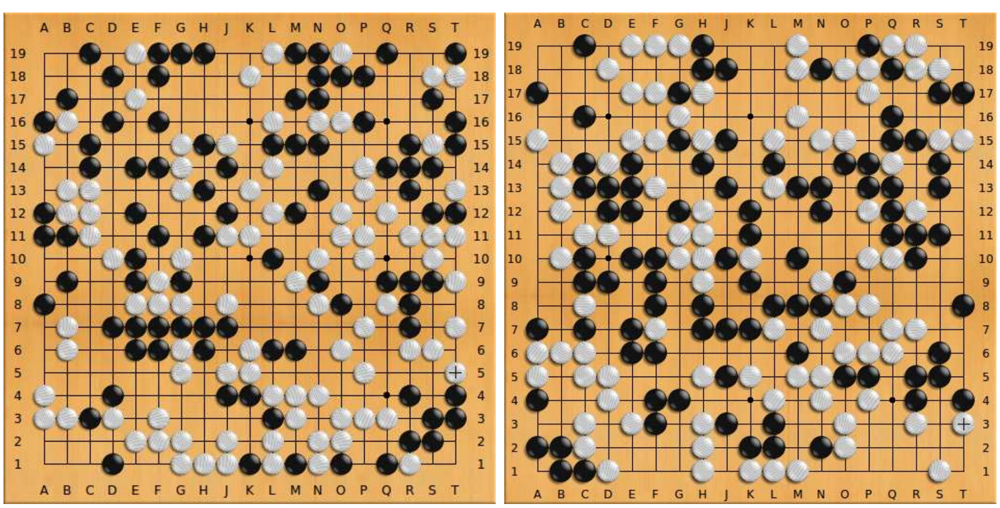

Fig. 11. Left: The first random position with 180 stones. The human chose white (believing that the situation was strongly in favor of white) and lost; then, he tried again as black and also lost. Results are presented in Fig. 12. Right: the second random position with 180 stones.

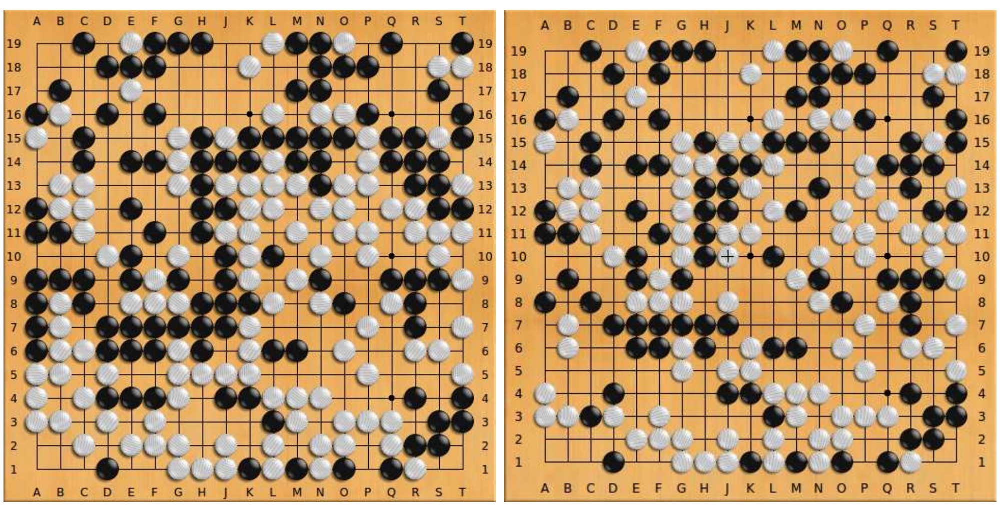

Fig. 12. The game won by the computer as black (left) and as white (right) from the first random position with 180 stones.